Tampereen ammattikorkeakoulu

Innovative hybrid marketing software

Teo Mäntymaa

BACHELOR'S THESIS
May 2019

IT
Software Engineering

Asiasanat: xamarin forms, tietotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikka
Ohjelmistotekniikka

Mäntymaa, Teo
Innovatiivinen hybridi-mainos-sovellus

Opinnäytetyö 24 sivua, joista liitteitä 0 sivua
Kesäkuu 2019

Fleurie on innovatiivinen mainossovellus, joka näyttää valitun alueen tai käyttäjää lähellä olevien kauppojen alennusprosentit merkkeinä kartalla. Esimerkki merkistä: Cafe Pispala –25%. Merkkiä painamalla saa lisätietoa tarjouksesta ja alennuskoodin, joka kuluttajan pitää tietää saadakseen alennuksen. Reitin kauppaan saa myös helposti painamalla yhtä nappia (Route). Reitti-toiminto käyttää Google Mapsia. Sovellus on toteutettu Xamarin Forms alustalla ja se toimii Android ja iOS järjestelmillä, eli on niinsanottu hybridi.

Sovelluksen avulla käyttäjät saavat edullisempia hintoja ja kaupat saavat tämän vuoksi enemmän kuluttajia, joka lisää myyntiä. Sovellus on suunnattu kivijalkakaupoille ja kaikille kaupan aloille. Sovelluksen markkina-aluetta ei ole rajattu Suomeen vaan se toimii myös ulkomailla.

ABSTRACT

Tampere University of Applied Sciences
Information technology
Software Engineering

Teo Mäntymaa
Innovative hybrid marketing software

Bachelor's thesis 24 pages, appendices 0 pages
June 2019

Fleurie is an innovative augmented reality marketing application that shows the discounts as markers on map. Example marker: Cafe Pispala -25%. Pressing the marker shows additional information about the deal and a unique discount code. Consumer must know the code in order to get the discount. Route to shop is easily obtained by pressing “Route"-button. Route function uses Google Maps. This software is built using Xamarin Forms platform. It is so called hybrid, because it runs on Android and iOS.

With this app, customers get better prices and shops get more customers. The application is free to use for customers, but shops must pay for their markers. The software is aimed for all fields of business and is not restricted to Finland only.

Key words: xamarin forms, information technology

SISÄLLYS

ABBREVIATIONS AND TERMS

Amazon ec2	Amazons Elastic Cloud Service
AWS	Amazon Web Service
Android	Googles Mobile Platform
C#	Programming Language by Microsoft
Xamarin	Platform for Mobile Operating Systems
MongoDB	Leading NoSQL Database
REST	Representation State Transfer
Slim	Rest Framework for PHP
iOS	Apples Mobile Platform
OSX	Apples Mac operating system
API	Application Programming Interface
SQL	Structured Query Language
NoSQL	Database that differs from traditional SQL relation db.
TTL	Time to live index, auto expiration of data.
Nginx	WWW and Proxy Server.
PHP	Programming Language
PHP-FPM	PHP Server
XAML	Extensible Application Markup Language
Swift	Apples Programming Language
MVVM	Model-View-ViewModel
IDE	Development Environment
Hybrid	Cross-platform

1 INTRODUCTION

Last summer on a sunny day at my front yard I was trying to think a good subject for this thesis project. This marketing software is what I came up with. It shows percentage discounts from shops as custom markers on augmented map.

The idea of using Xamarin Forms framework (cross platform) came from Kennelliitto. They said to me that only Android is not enough when cross-platform solutions exist. I did the research and Xamarin Forms seemed like the best option because the apps are native and c# is a win over Apples swift language.

Front-end uses Microsoft's Xamarin Forms platform. UI's are built with XAML. Large portion of code is shared between platforms (Android and iOS).

Back-end is hosted on cloud (Amazon ec2). It uses Ubuntu Linux. Good thing about clouds is that you get static ip address. Web server of choice is Nginx and PHP-fpm for php support. Remote database used is MongoDB.

Marketing and selling markers is the hardest part of this project. I was thinking about making a flyer that can be spread to shop owners. I also have an offline website that can sell markers.

2 MAIN FEATURES

2.1 Front-end

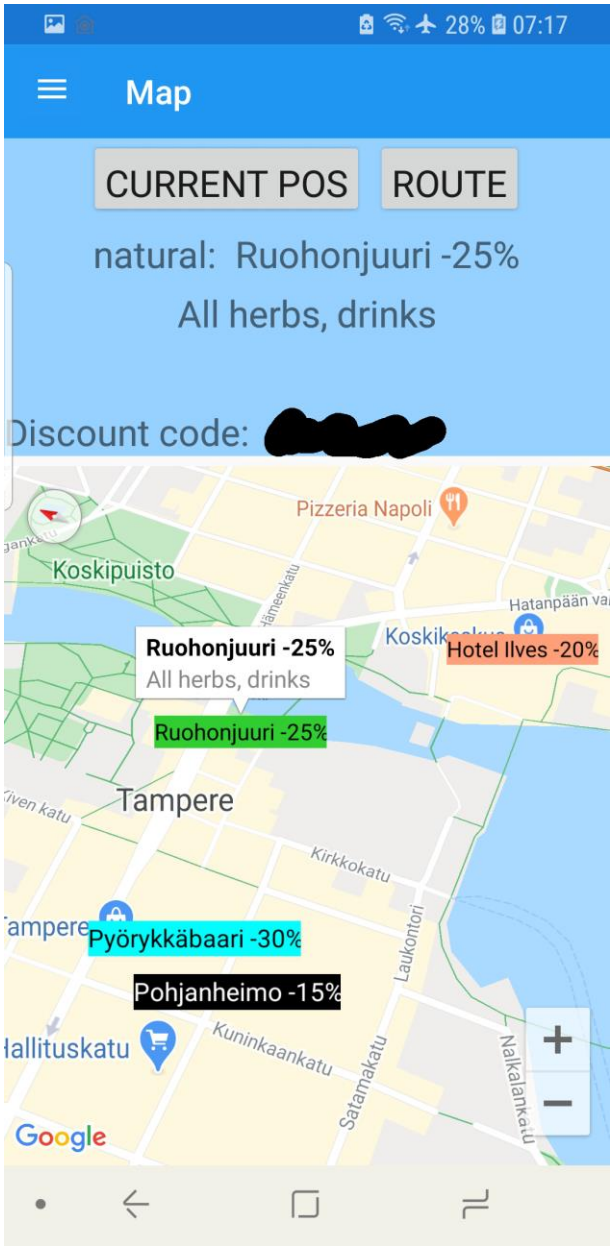
Front-end uses Xamarin Forms. It is a cross platform framework for embedded systems (Android and iOS). This is a master – detail application. Pressing hamburger-menu opens the master list. This front-end uses the MVVM-model.

Views / UI's are built using XAML and a code behind class. Views are tested to work on pads and mobiles. Apple Developer program is a requirement for running your own software on IOS and it currently costs 100e / year.

2.1.1 Map View

This is the main view. It is a Google Maps view augmented with custom markers. The markers represent discounted deals. Pressing a marker gives additional information and discount code that is required for the deal. Each category of markers has its own color. Shown categories can be altered from Settings-view.

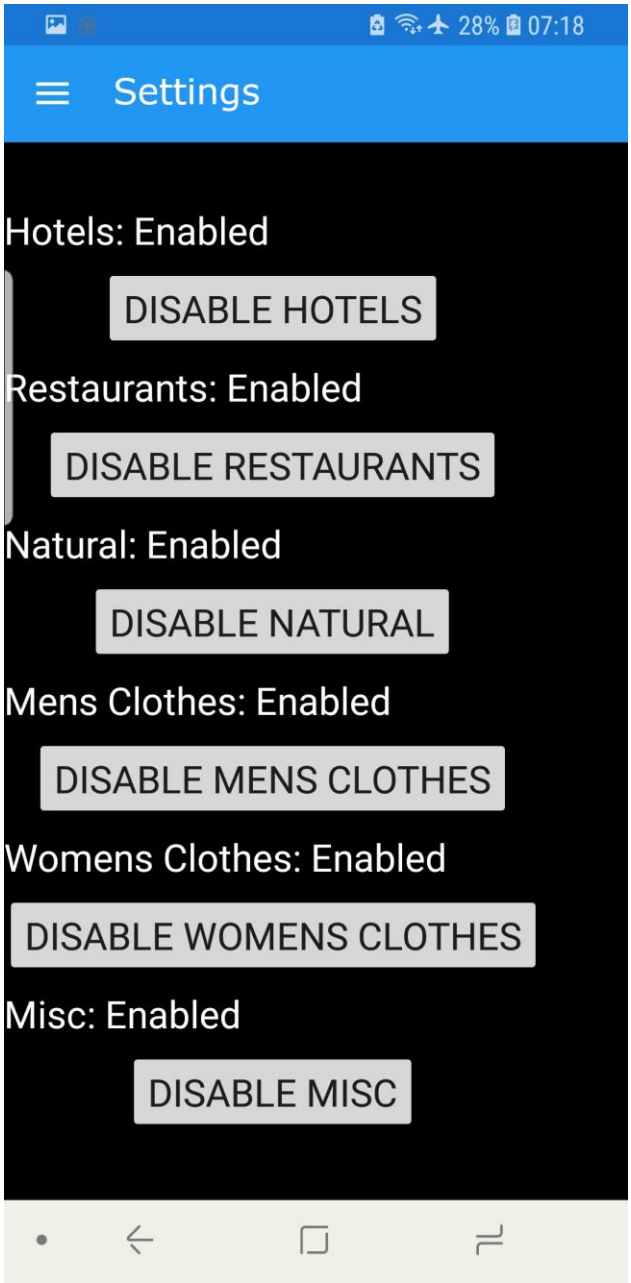
"Current pos"-button zeroes the map to users current coordinates. "Route"-button launches Google Maps app for easy route directions. Internet connection and location permission are required for this view to load.



PICTURE 1. Main map-view

2.1.2 Settings View

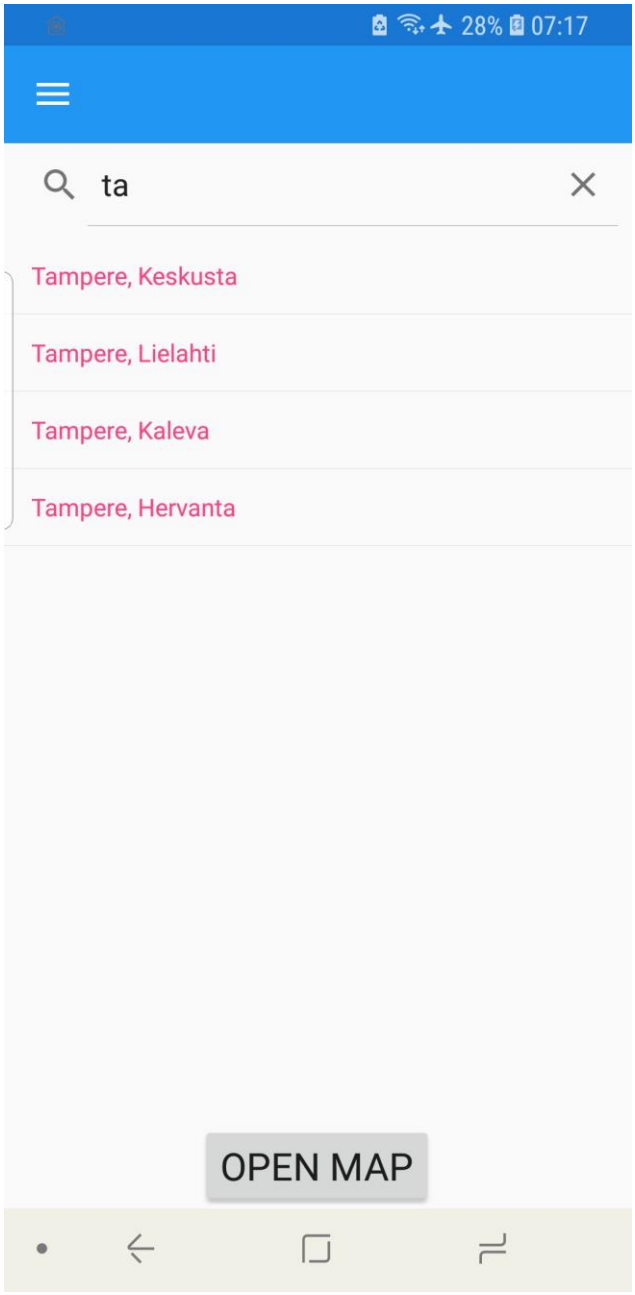
Settings for showing or hiding different genres on main map.



PICTURE 2. Settings view

2.1.3 Change Location - View

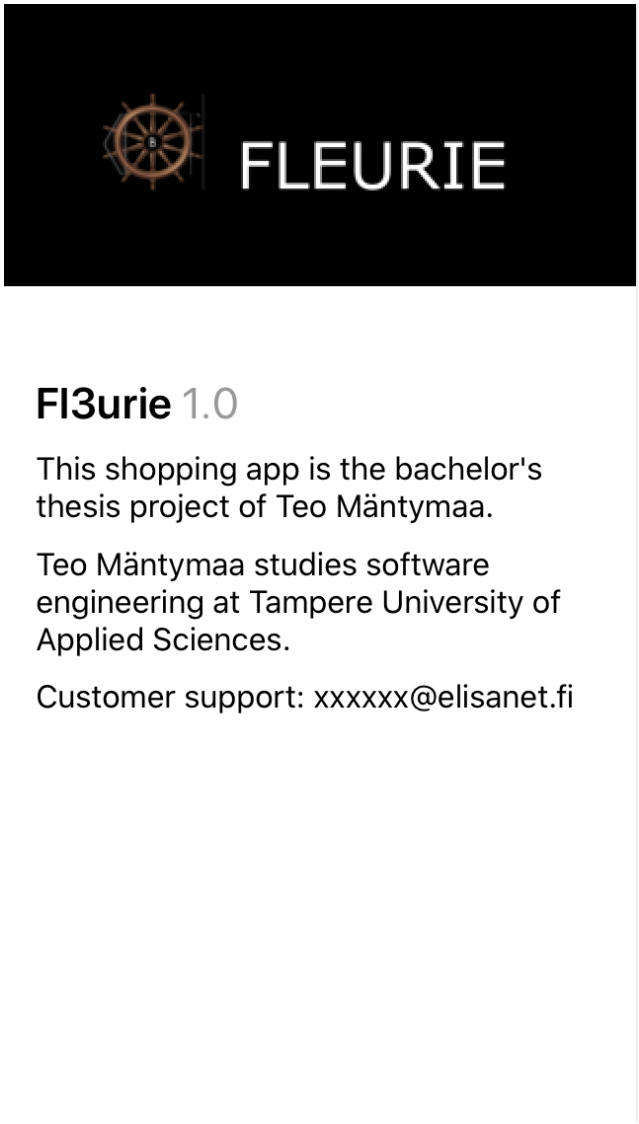
This view allows changing your location to different city or area. It features a search bar for filtering areas from local SQLite database.



PICTURE 3. Change location view

2.1.4 About View

This view has some background and contact infomation.



PICTURE 4. About view

2.2 Back-end

Back-end is hosted on Amazon Elastic Cloud 2. Ec2 T2.micro instance is free for 12 months at the moment (2020). Server uses Ubuntu Linux for its remarked stability. It supports MongoDB database and PHP API (Slim). Server runs Nginx + PHP-FPM web-server. The server maps HTTP-request routes to Mongo-queries and returns matching results as map-markers.

There are other options to consider like Google Cloud or Microsoft Azure. In-depth comparison is written by Soinu Anssi (2014). One good feature of using cloud services is that you get static ip address, witch is a requirement.

How to connect to your cloud using command line tool in OSX:

Replace XXXXX with your own server info:

ssh -i "Fleurie_pem_key.pem" ubuntu@ec2-XXXXXXXXX.us-east-2.compute.amazonaws.com

2.2.1 MongoDB Database

MongoDB is the leading NoSQL database at the moment (year 2020). Mongo supports geospatial queries meaning it can return all the markers below certain distance.

Another good feature is the TTL index doing automatic data expiration. This allows selling markers for a specific time period.

2.2.2 Nginx web server

Ningx with php-fpm is a high performance PHP web server used to host the HTTP API. Some configuration tweaking is required with this server. Https is a requirement for Android past API 27+.

Enable https (Justin Ellingwood, 2014). First create a folder :

sudo mkdir /etc/nginx/ssl

Then create sertificate:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/nginx.key -out /etc/nginx/ssl/nginx.crt

Lastly insert these three orange lines in your Nginx server config block:

```
server {
        listen 80 default_server;
        listen [::]:80 default_server ipv6only=on;

        listen 443 ssl;

        root /usr/share/nginx/html;
        index index.html index.htm;

        server_name your_domain.com;
        ssl_certificate /etc/nginx/ssl/nginx.crt;
        ssl_certificate_key /etc/nginx/ssl/nginx.key;

        location / {
                try_files $uri $uri/ =404;
        }
}
```

PICTURE 5. Enabling https

2.2.3 HTTP API (PHP)

Slim framework is used for routing, Mongo has good support for PHP.

Example route: https://3.16.180.88/index.php/get/23.6339/60.8146

```php
<?php
header('Content-Type: application/json');
header("Access-Control-Allow-Origin: *");

require dirname(__FILE__).'/vendor/autoload.php';

use \Slim\Slim;

Slim::registerAutoloader();

//Instantiate a Slim application:
$app = new Slim();

//Define a HTTP GET route:
$app->get('/', function () {
    echo "Hello!";
});

$app->get('/hello/:name/', function ($name) {
    echo "Hello, $name";
});

$app->get('/get/:lat/:lon/', function ($lat, $lon) {
//    echo "Hello, $lat";

$flon = floatval($lon);
$flat = floatval($lat);

$collection = (new MongoDB\Client)->fin->fin;

$cursor = $collection->find(Array("loc" => Array('$geoWithin' => Array('$centerSphere'  => Array(Array($flon, $flat),
20.0/6378)))));

$json = json_encode(iterator_to_array($cursor), JSON_FORCE_OBJECT);
echo $json;

//foreach ($cursor as $document) {
 //   echo $document['tag'], "\n";
//}

});


//Run the Slim application:
$app->run();
?>
```

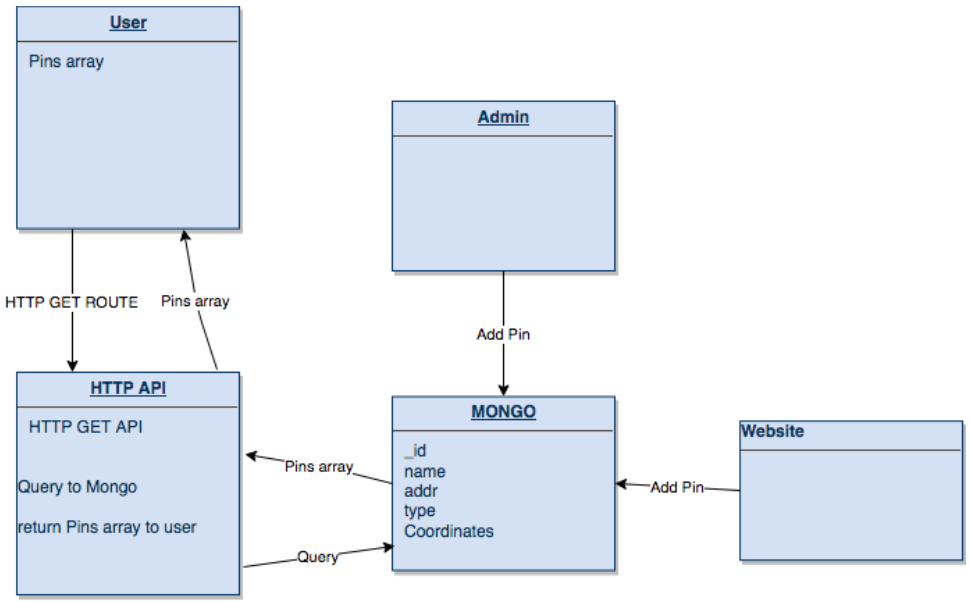
PICTURE 6. index.php and GET-routes

3 ARCHITECTURE

IDE used is Visual Studio for Mac. This project uses Git (gitlab.tamk.cloud) to host source files.

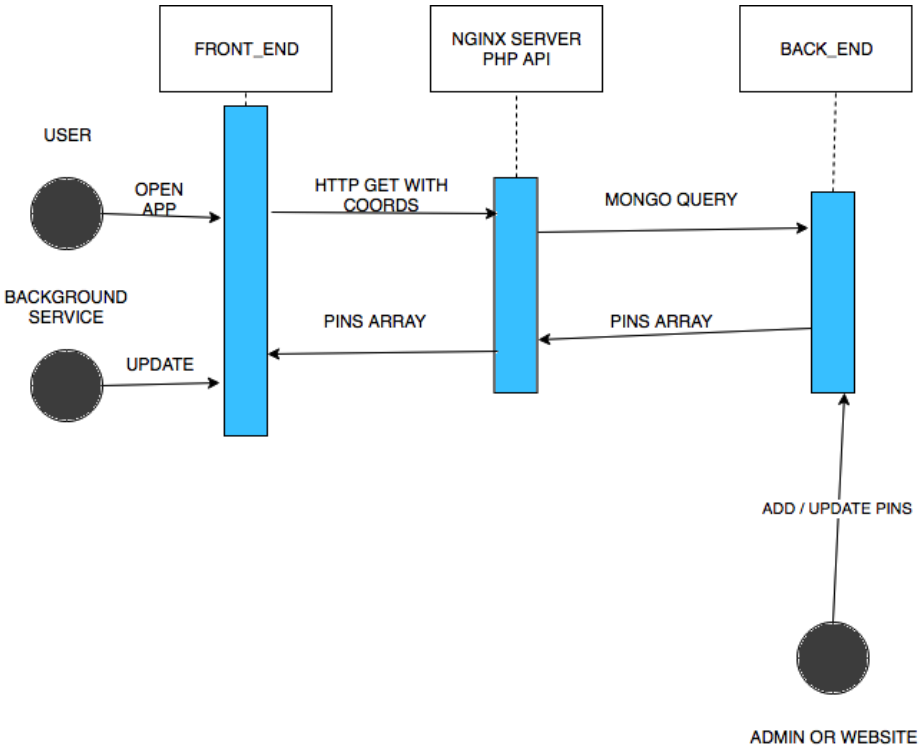
The architecture is fairly simple so I won't go overboard with diagrams. UML Class diagram and sequence diagram will do.

3.1 Class diagram



Picture 7. Class diagram of the project

3.2 Sequence diagram



Picture 8. Sequence diagram of the project

4 CODE

4.1 Basics

4.1.1 X:Name

```
The id's/names of UI components are set using x:Name="xy" in XAML:

<Label Text="Disable Restaurants" x:Name="restaurants_label" HorizontalOptions="Center" />

Example of x:Name usage in code behind class:

restaurants_label.Text = "Frodo";
```

Picture 9. x:Name

4.1.2 Button

The OnClick method for button is done by first setting the x:Name in XAML:

```
<Button Text="Disable Restaurants" x:Name="restaurants_button" HorizontalOptions="Center" />
```

Implement restaurants_button.Clicked lambda function in code dehind:

```
restaurants_button.Clicked += (sender, e) =>
{
        if (((Button)sender).Text == "Disable Restaurants")
        {
            restaurants_label.Text = "Restaurants: Disabled";
        }
        else
        {
            restaurants_label.Text = "Restaurants: Enabled";
        }
  };
```

Picture 10. Button clicked

4.1.3 Shared Preferences

Shared preferences are a way of storing small amounts of data in a text file.

```
Set shared preference:
        Preferences.Set("latitude", 10.0d);
Get shared preference:
        var myLat = Preferences.Get("latitude", defaultVal);
```

Picture 11. Shared preferences usage

4.1.4 Toast

Toast is a way of showing string overlay messages. Add Plugin.Toast Nuget by ish-rakland to all 3 projects.

Plugin is easy to use and supports custom colors (Ishrakland, 2020).

```
Basic usage:

    CrossToastPopUp.Current.ShowToastMessage("Fleurie");

Custom colors:

    CrossToastPopUp.Current.ShowCustomToast("Fleurie", Color.Aqua.ToHex(), "#4d4c49",
    (ToastLength)"Fleurie".Length);
```

Picture 11. Toast usage

4.1.5 GoogleMap

I followed this guide to set up GoogleMaps for Android:

https://docs.microsoft.com/en-us/xamarin/android/platform/maps-and-location/maps/obtaining-a-google-maps-api-key?tabs=macos

Setup shared map between Android and iOS:
Install Xamarin.Forms.Googlemaps nugget to all 3 projects. iOS needs also Xamarin.Build.Download nugget.

Follow instructions here:
https://github.com/amay077/Xamarin.Forms.GoogleMaps

Starting on iOS 8.0 and above, you must set NSLocationWhenInUseUsageDescription or NSLocationAlwaysUsageDescription in info.plist. This will enable location services.

4.2 Shared interface

Separate function implementations for Android and iOS using interface (Vimal.4745, 2016).

Put in your shared view/Page:

```
namespace Fl3urie.Views
{
    public partial class MapPage : ContentPage
    {
        //The shared interface
        public interface CalculateTextWidth { double calculateWidth(string text); }

        public void setPins()
        {
        //Using the shared calculateWidth() function that is on the interface
        double y = DependencyService.Get<CalculateTextWidth>().calculateWidth(x.cNameAddAmount);
        }
    }
}
```

Picture 12. Shared part of separate implementation

Put in Android MainActivity.cs:

```
using static Fl3urie.Views.MapPage;

[assembly: Dependency(typeof(Fl3urie.Droid.CalculateTextWidth_Android))]. //Important

namespace Fl3urie.Droid
{
        public class CalculateTextWidth_Android : CalculateTextWidth
         {
        public CalculateTextWidth_Android() { }
        public double calculateWidth(string text)
        {
            Rect bounds = new Rect();
            TextView textView = new TextView(Forms.Context);
            textView.Paint.GetTextBounds(text, 0, text.Length, bounds);
            var length = bounds.Width();
            return length / Android.Content.Res.Resources.System.DisplayMetrics.ScaledDensity;
        }
    }
public class MainActivity …
```

Picture 13. Android part of separate implementation

Put in iOS Main.cs:

```
using static Fl3urie.Views.MapPage;

[assembly: Dependency(typeof(Fl3urie.iOS.CalculateTextWidth_iOS))] //Important

namespace Fl3urie.iOS
{
    public class CalculateTextWidth_iOS : CalculateTextWidth
    {
        UILabel uiLabel;
        CGSize length;
        public CalculateTextWidth_iOS()
        {
        }

        public double calculateWidth(string text)
        {
            uiLabel = new UILabel();
            uiLabel.Text = text;
            length = uiLabel.Text.StringSize(uiLabel.Font);
            return length.Width;
        }
    }
    public class Application
    { …
```

Picture 14. iOS part of separate implementation

4.3 HttpClient

This client sends http request to my php + mongo database api and then consumes the response. All the methods of this class are async so they won't stall your ui thread. Android API level 27+ requires https web-api.

```
List<JToken> myPinListRemote { get; set; }


 async public void getFromRemote()
{

     var request = new HttpRequestMessage();


     string u = "";

     u = "https://3.16.180.88/index.php/get/" + lon.ToString() + "/" + lat.ToString();   //Android use https

     if (Device.OS == TargetPlatform.iOS) {
            u = "http://3.16.180.88/index.php/get/" + lon.ToString() + "/" + lat.ToString();    //iOS use http
     }

     request.RequestUri = new Uri(u);

     request.Method = HttpMethod.Get;
     request.Headers.Add("Accept", "application/json");

     var handler = new HttpClientHandler();

     handler.ServerCertificateCustomValidationCallback = (sender, cert, chain, sslPolicyErrors) => { return true; };

     var client = new HttpClient(handler);

     HttpResponseMessage response = await client.SendAsync(request);

     if (!(response.StatusCode == System.Net.HttpStatusCode.OK))
     {
     // CrossToastPopUp.Current.ShowToastMessage(response.ToString());
    return;
     }

     //parse JSON and grab children.

     string response2 = await response.Content.ReadAsStringAsync();

     var jsonData = JObject.Parse(response2).Children();

     myPinListRemote = jsonData.Children().ToList();
 }
```

Picture 15. HttpClient usage

5 CONCLUSION AND LAST WORDS

This is my first cross-platform software project for embedded systems. I learned a lot new technical stuff from this project and the goals were met. Selling markers is the hardest part of this project but the market for this type of product must exist since marketing is such a large field of business.

Im looking forward to do more projects using the same frameworks.

REFERENCE LIST

Justin Ellingwood. 2014. How To Create an SSL Certificate on Nginx for Ubuntu 14.04. Read 24.04.2020. https://www.digitalocean.com/community/tutorials/how-to-create-an-ssl-certificate-on-nginx-for-ubuntu-14-04

Vimal.4745. 2016. How to calculate or measure width os a string. Read 11.03.2020. https://forums.xamarin.com/discussion/67545/how-to-calculate-or-measure-width-of-a-string

Amay077. Xamarin.Forms.GoogleMaps. Read 12.03.2020. https://github.com/amay077/Xamarin.Forms.GoogleMaps

Microsoft. Obtaining a Google Maps API Key. Read 14.03.2020. https://docs.microsoft.com/en-us/xamarin/android/platform/maps-and-location/maps/obtaining-a-google-maps-api-key?tabs=macos

Ishrakland. Toast. Read 07.03.2020. https://github.com/ishrakland/Toast

Soinu, Anssi. 2014. Cloud Solutions for Mobile Applications. Information technology. Tampere University of Applied Scienses. Master's thesis.